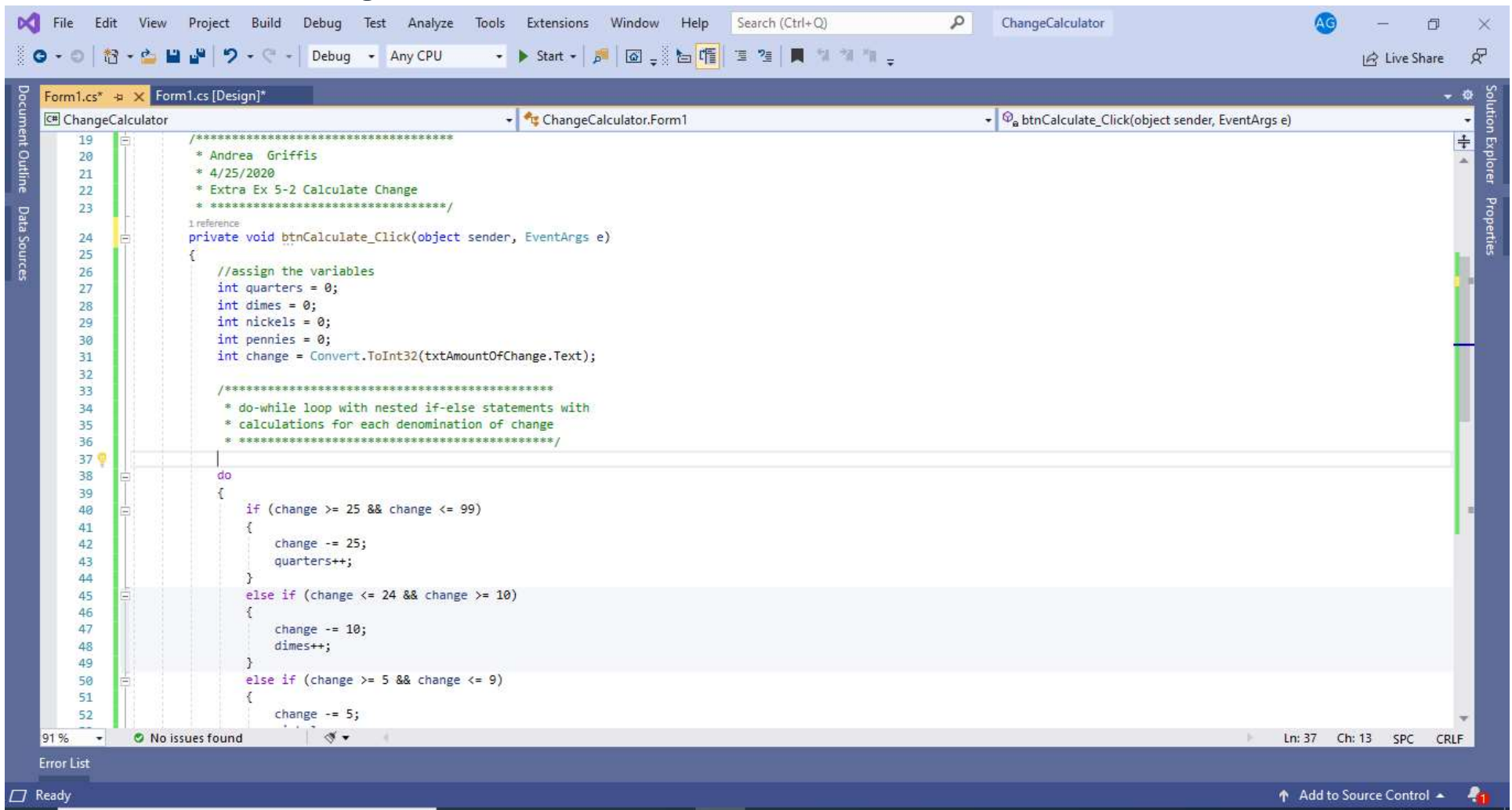


## Assignment: Week 3 Murach Coding Assignments (Individual)

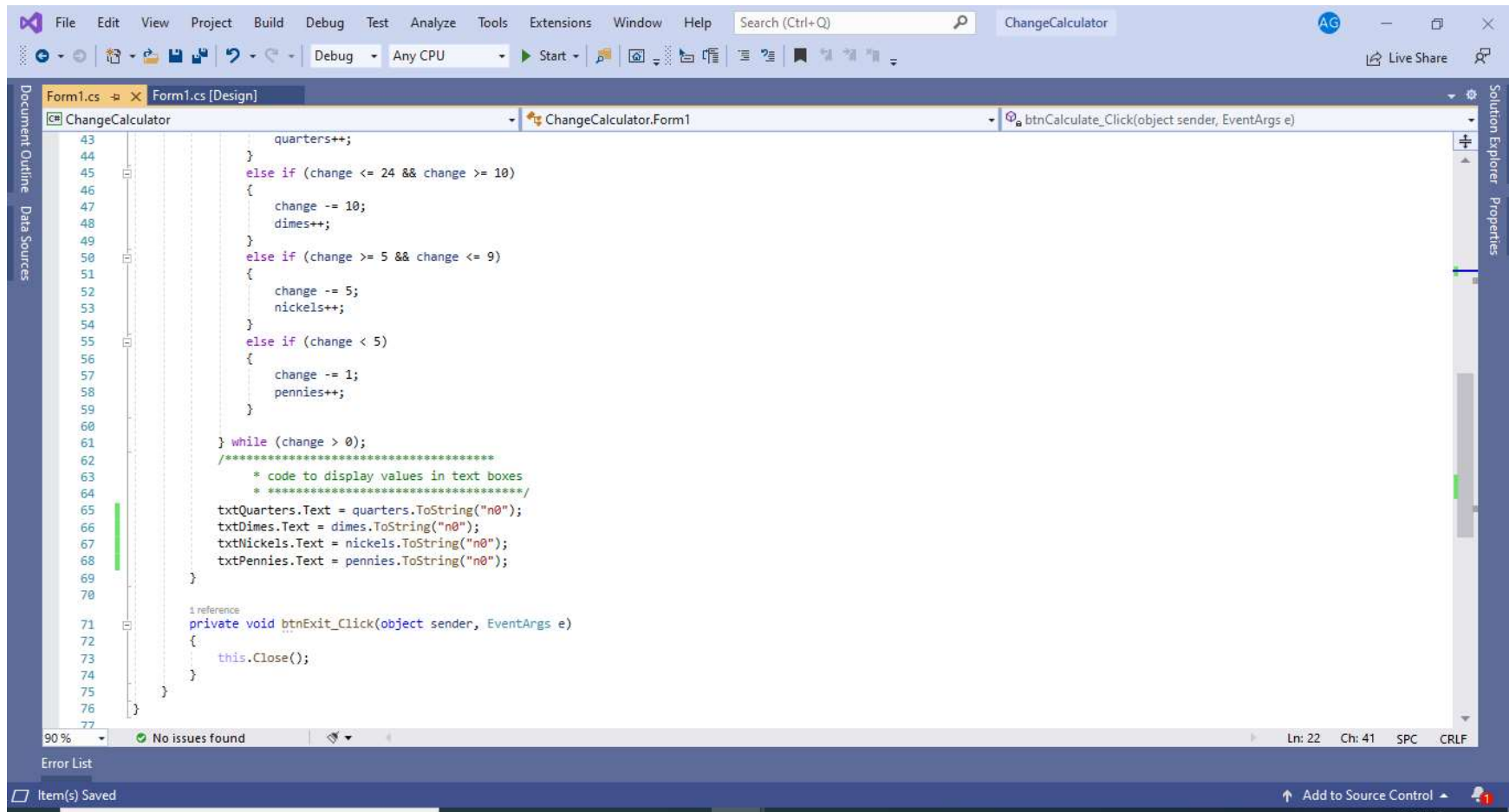
## Extra Exercise 5-2 Calculate Change



The screenshot shows the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q)
- Toolbar:** Includes icons for Run, Stop, Break, and other debugging tools.
- Document Explorer:** Shows the project structure with 'Form1.cs' and 'Form1.cs [Design]\*'.
- Code Editor:** Displays the C# code for 'btnCalculate\_Click(object sender, EventArgs e)'. The code includes comments and logic for calculating change using a do-while loop and nested if-else statements.
- Solution Explorer:** Shows the project structure with 'ChangeCalculator' and 'ChangeCalculator.Form1'.
- Properties Window:** Shows the properties of the selected element.
- Status Bar:** Displays '91 %', 'No issues found', and 'Ln: 37 Ch: 13 SPC CRLF'.

```
19  /*****
20  * Andrea Griffis
21  * 4/25/2020
22  * Extra Ex 5-2 Calculate Change
23  * *****/
24  1 reference
25  private void btnCalculate_Click(object sender, EventArgs e)
26  {
27      //assign the variables
28      int quarters = 0;
29      int dimes = 0;
30      int nickels = 0;
31      int pennies = 0;
32      int change = Convert.ToInt32(txtAmountOfChange.Text);
33
34      /*****
35      * do-while loop with nested if-else statements with
36      * calculations for each denomination of change
37      * *****/
38      do
39      {
40          if (change >= 25 && change <= 99)
41          {
42              change -= 25;
43              quarters++;
44          }
45          else if (change <= 24 && change >= 10)
46          {
47              change -= 10;
48              dimes++;
49          }
50          else if (change >= 5 && change <= 9)
51          {
52              change -= 5;
```



Change Calculator

Amount if change due (0-99):

Quarters:

Dimes:

Nickels:

Pennies:

Change Calculator

Amount if change due (0-99):

Quarters:

Dimes:

Nickels:

Pennies:

Change Calculator

Amount if change due (0-99):

Quarters:

Dimes:

Nickels:

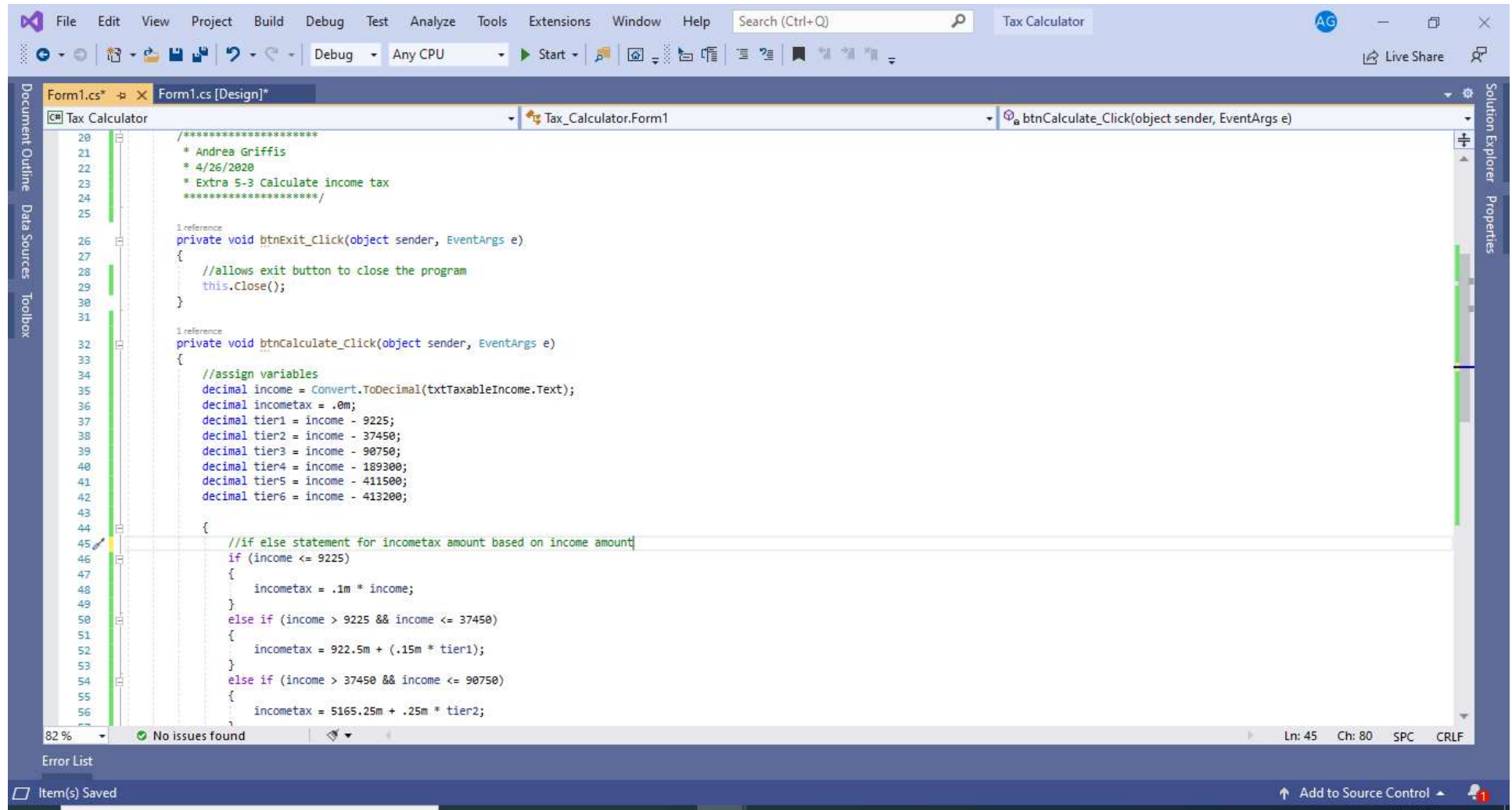
Pennies:

## Extra Exercise 5-3 Calculate Income Tax

VIDEO>> <http://youtu.be/AueWbSV-1Tw?hd=1>

I was a little stuck on this one as the example in the exercise says to use 8700 and 35350 then it will bring you 870 and 4840.50. However, the calculations states that the income tax for and income of 35350 is 15% on the amount over 9225 plus 922.50. According to my calculations below it should be 4841.25 not 4840.50

$35350 - 9225 = 26125$ ; 15% of 26125 is 3918.75;  $3918.75 + 922.50 = 4841.25$ , I'm not sure how to get 4840.50



```
20  /*****  
21  * Andrea Griffis  
22  * 4/26/2020  
23  * Extra 5-3 Calculate income tax  
24  *****/  
25  
26  1 reference  
27  private void btnExit_Click(object sender, EventArgs e)  
28  {  
29      //allows exit button to close the program  
30      this.Close();  
31  }  
32  
33  1 reference  
34  private void btnCalculate_Click(object sender, EventArgs e)  
35  {  
36      //assign variables  
37      decimal income = Convert.ToDecimal(txtTaxableIncome.Text);  
38      decimal incometax = .0m;  
39      decimal tier1 = income - 9225;  
40      decimal tier2 = income - 37450;  
41      decimal tier3 = income - 90750;  
42      decimal tier4 = income - 189300;  
43      decimal tier5 = income - 411500;  
44      decimal tier6 = income - 413200;  
45  
46      //if else statement for incometax amount based on income amount  
47      if (income <= 9225)  
48      {  
49          incometax = .1m * income;  
50      }  
51      else if (income > 9225 && income <= 37450)  
52      {  
53          incometax = 922.5m + (.15m * tier1);  
54      }  
55      else if (income > 37450 && income <= 90750)  
56      {  
57          incometax = 5165.25m + .25m * tier2;  
58      }  
59  }  
60  }
```

Visual Studio Code interface showing the Tax Calculator application. The main window displays the C# code for the `btnCalculate_Click` event handler in `Tax_Calculator.Form1`. The code implements a tiered income tax calculation based on the taxable income amount.

```
//if else statement for incometax amount based on income amount
if (income <= 9225)
{
    incometax = .1m * income;
}
else if (income > 9225 && income <= 37450)
{
    incometax = 922.5m + (.15m * tier1);
}
else if (income > 37450 && income <= 90750)
{
    incometax = 5165.25m + .25m * tier2;
}
else if (income > 90750 && income <= 189300)
{
    incometax = 18481.25m + .28m * tier3;
}
else if (income > 189300 && income <= 411500)
{
    incometax = 46075.25m + .33m * tier4;
}
else if (income > 411500 && income <= 413200)
{
    incometax = 119996.25m + .35m * tier5;
}
else if (income > 413200)
{
    incometax = 119996.25m + .396m * tier6;
}
decimal taxowed = incometax;
txtIncomeTaxowed.Text = taxowed.ToString("c");
```

The status bar indicates 82% zoom, no issues found, and the cursor is at line 45, column 80.

Two screenshots of the **Income Tax Calculator** application window are shown side-by-side.

**Left Screenshot:**

- Taxable income: 8700
- Income tax owed: \$870.00
- Buttons: Calculate, Exit

**Right Screenshot:**

- Taxable income: 35350
- Income tax owed: \$4,841.25
- Buttons: Calculate, Exit